Appendix

The following shows examples of instructions automatically generated to answer different types of grammatically-context-free questions using the embodiment shown in FIG. 2.

```
1.    Who is the first President?
for Table in each Tables-Of(President)
do
        for Attribute1 in Attribute-Names(President, Table)
        do
                for Attribute2 in Attribute-Names(first, Table)
                do
                        res = (SELECT DISTINCT Attribute-Name(who, Table)
                                FROM Table
                                WHERE Attribute1 = "President"
                                ORDER BY Attribute2 ASC)
                        if (res is not empty) return {first element of results}
                end for
        end for
end for
return {error, no solution found}

2.    What are the Bills of Right?
answer = ""
for Table in each Tables-Of("Bills of Right")
do
        for Key in Keys-Of(Table)
        do
                x = (SELECT Attribute-Name(what, Table) FROM Table
                        WHERE Key LIKE 'Bills of Right');
```

```
                    answer = answer + x
                endfor
            endfor
            if answer is not empty, return answer, otherwise return error.
```

3. Where is the capital of Texas?

There are two nouns in this question: capital and Texas. In one embodiment, each noun is associated with a topic-related table. The preposition "of" indicates that the table associated with Texas should take precedence over the capital table.

This question can have two different answers. The answer can be the geographic location as shown on a map, or the answer can be the city Austin. One embodiment generates the following instructions, with x denoting Austin, and y denoting geographic location:

```
for Table in each Tables-Of(Texas)
do
        for Key in Keys-Of(Table)
        do
                for Attribute in Attribute-Names(capital, Table)
                do
                        x = (SELECT DISTINCT Attribute
                                FROM Table
                                WHERE Key LIKE Texas)
                        if (x is valid) then goto found
                end for
        end for
end for
if {x is not valid) or (x has more than one value) then return error
found :
for Table in Tables-Of(x)
do
```

```
                for Key in Keys-Of(Table)
                do
                        y = (SELECT DISTINCT Attribute-Name(where, Table)
                                FROM Table
                                WHERE Key LIKE x)
                        if (x is valid) then goto found
                end for
        end for
        return {error, no solution found}
```

4. What is the immediate cause of the Civil War?

In this example, the word "immediate" when linked with the noun "cause" is equivalent in meaning to the word "last."

```
        for Table in each Tables-Of("Civil War")
        do
                for key in Keys-Of(Table)
                do
                        for Attribute1 in Attribute-Names(cause, Table)
                        do
                                for Attribute2 in Attribute-Names(last, Table)
                                do
                                        x = (SELECT Attribute1 FROM Table
                                                WHERE Key LIKE "Civil War"
                                                ORDER BY Attribute2 DESC)
                                        if (x is valid) then return {first element in x}
                                end for
                        end for
                end for
        end for
```

5. Why did President Nixon resign?

Events about "President Nixon resign" are extracted into a local table T. Then the causes of all such events are identified.

```
CREATE LOCAL TABLE T AS
        (SELECT  * FROM Verb_View(resign)
                WHERE subject-agent = "President Nixon");
SELECT A.description
        FROM EVENT A, T, EVENT-RELATIONSHIP R
        WHERE R.KeyId1 = A.KeyId
            AND R.KeyId2 = T.KeyId
            AND R.relationship = "consequence";
```

The following examples 6-9 depend on tables, such as the President tables, generated above.

6. Who is the third President?

```
names = (SELECT DISTINCT Name FROM PRESIDENT
        ORDER BY Start_Year ASC)
```

return the 3rd name.

These instructions assume Tables-Of(President) contains only one table, and Attribute-Names('third', PRESIDENT) yields only a single attribute.

7. Who is the President after John Kennedy?

```
SELECT DISTINCT Name FROM PRESIDENT
        WHERE Start_Year > (SELECT Start_Year FROM
        PRESIDENT WHERE Name LIKE "John Kennedy")
ORDER BY Start_Year DESC;
```

select the first result.

8. When did President Lyndon Johnson die?

SELECT Death_Date FROM PRESIDENT WHERE Name LIKE "Lyndon Johnson";

9. When was President Nixon born?

SELECT Birth_Date FROM PRESIDENT WHERE Name LIKE "Nixon"

A number of mathematical examples are shown in the following using the programming language Lisp. Some examples might use mathematical software packages.

For mathematics problems, there can be an additional topic-related table, known as the mathematics table. The mathematics table might include high-level attributes, such as fraction, ratio, derivative, theorem, complex-number, matrix etc. Each of these high-level attributes can have a definition attribute describing the mathematical concept; a reference attribute identifying the study material covering the concept; and an algorithm attribute, if there is one. In another embodiment, the algorithm attribute references a mathematical software package. A high-level attribute can be a theorem, such as the Mean Value Theorem. In its corresponding definition attribute, there will be an explanation for that theorem. Note that, in this example, the algorithm attribute is not in the topic-dependent semantic table.

Most mathematical questions are 'what-is' questions. If a question is of the form 'What is X?', where X is a mathematical term, the system generates the following instruction:

SELECT definition FROM T mathematics table WHERE T.name = X;

If the question is 'What is X <prepositional-noun-phrase>?', the <prepositional-noun-phrase> modifies the term X. The system retrieves X and applies the operations to the terms indicated by the prepositions.

10. What is the ratio between 7 and 8?

From the mathematics table, the attribute "ratio" is identified. Its algorithm attribute operates on two terms, as specified in the prepositional-phrase "between 7 and 8." For example:

(ratio 7 8)

11. What is the derivative of sin(x^2 + 4) with respect to x?

Again, the question is of the structure: "What is X <prepositional-noun-phrase>?" X is "the derivative," and the noun-phrase is "sin(x+4) with respect to x." The algorithm specified by the "derivative attribute" in the mathematics table has two operants, u and v, with the first operant, u, being the noun immediately after the preposition. In this example, u is equal to sin(x^2+4), and the second operant is v. The system can invoke a mathematics software package to solve the differentiation, d(sin(x+4))/dx. One LISP representation is as follows:

(derivative 'x '(sin (+ (* x x) 4)))

12. What is the product of {matrix-1} and {matrix-2}?

The LISP representation is as follows:

(matrix-mul <matrix-1> <matrix-2>)

13. What is the sum of the product of 3 and 4, and the ratio of 7 and 8?

The LISP representation is as follows:

(+ (* 3 4) (ratio 7 8))

Not all mathematics questions are 'What-is' questions. The user may not understand certain steps in a derivation he is studying on the screen. Such questions are typically 'Why-is' questions. In one embodiment, the system identifies the study material, as in the current screen or the last screen he has been focusing on. Nouns used in the question are mostly from those study material. By associating the question with the study material and accessing a database of explanation terms in the study material, the system can generate an answer to the question.

14. Why is delta used in step 4 of the proof?

The structure of the question is: Why is <noun phrase> <verb phrase>. The system

(a) Identify the verb in the verb phrase to be the word "used."

(b) Identify one or more sets of events algorithm corresponding to the word "used" in the mathematics table. The sets of algorithm are applicable to the structures of X using Y.

(c) Identify X to be (delta).

(d) Identify Y to be (in step 4 of the proof).

(e) Identify the i-pronoun and the aux-verb as "Why is."

(f) Select the set of algorithm for explaining the relationship between X and Y. In this case, the set selected explains why X used Y.

(g) The algorithm searches for delta in step 4 of the proof in the materials just presented to the user. After identifying the proof, step 4 and delta, the system explains the step 4 of the proof with special focus on delta.